

JavaScript Bewijslast

(Een aantal voorbeelden van toepassingen in JavaScript)

Variabelen

```
const player = document.getElementById('player');
const gameContainer = document.querySelector('.gamecontainer');
const invadersContainer = document.getElementById('invaders');
const scoreElement = document.getElementById('score');

let PLAYERPOSITION = 275; // start positie van de player
let BULLETARRAY = []; // lege array die de kogels van de speler opslaat
let INVADERSARRAY = [];
let INVADERSDIRECTION = 10;
let GAMEINTERVAL;
let lastShotTime = 0; // voorkomt dat er niet te snel achter elkaar geschoten wordt
const SHOT_DELAY = 500; // aantal ms dat je moet wachten per bullet
let level = 1; // spelniveau, elk level = 5 nieuwe vijanden
let score = 0;
```

(Eggstermination)

```
var character = document.getElementById("character")
var block = document.getElementById("block")
var GAMEOVER = document.getElementById("GAMEOVER")
var scoreElement = document.getElementById("score");
var highScoreElement = document.getElementById("highscore");
var isGameOver = false; // checkt of het Game Over is, staat dus uit
var score = 0; // huidige score, staat natuurlijk op 0 als de game begint
var highScore = localStorage.getItem('highScore') || 0; // localStorage.getItem('highScore')
// als je de website afsluit en later weer opstart, blijft de high score behouden
highScoreElement.innerHTML = "High Score: " + highScore; // zorgt ervoor dat de high score wordt weergegeven
var jumpSound = document.getElementById("jumpSound");
jumpSound.volume = 0.1;
var GAMEOVERSound = document.getElementById("GAMEOVERSound");
GAMEOVERSound.volume = 0.1;
```

(T-Rex Escape)

```
const columnRanges = [
  { start: 1, end: 15 },
  { start: 16, end: 30 },
  { start: 31, end: 45 },
  { start: 46, end: 60 },
  { start: 61, end: 75 }
];
```

(Bingo)

Comments

```
var highScoreElement = document.getElementById('highScore');
var isGameOver = false; // checkt of het Game Over is, staat dus uit als de game net wordt opgestart
var score = 0; // huidige score, staat natuurlijk op 0 als de game begint
var highScore = localStorage.getItem('highScore') || 0; // localStorage = opgeslagen high score.
// als je de website afsluit en later weer opstart, blijft de high score ^
highScoreElement.innerText = "High Score: " + highScore; // zorgt ervoor dat de high score wordt gedisplayd.
var jumpSound = document.getElementById("jumpSound");
jumpSound.volume = 0.1;
var GAMEOVERSound = document.getElementById("GAMEOVERSound");
GAMEOVERSound.volume = 0.1;

document.addEventListener('keydown', function(event) { // checkt of er een knop wordt ingedrukt
  if (event.code === 'Space') { // checkt of de spatiebalk wordt ingedrukt

    score = -1; // de score begint op -1, want als je op de Game Over knop klikt telt deze klik ook als een jump
    // hierdoor zal deze klik ervoor zorgen dat de score recht voor de game begint van -1 naar 0 switched
    // de player ziet dit niet gebeuren en zal alleen merken dat de game gewoon op een score van 0 begint
    scoreElement.innerText = "Score " + score;
    block.style.animation = "block 1s infinite linear"; // start de animatie en loop van de blocken (eieren)
    block.style.display = "block";
    GAMEOVER.style.display = "none";
    character.style.top = "600px"; // reset de positie van de player
    block.style.left = "1480px"; // reset de positie van de eieren
    checkDead = setInterval(function() { // zie uitleg van checkDead in het vorige paragraaf
```

(T-Rex Escape)

```
// functie die de players bullets laat schieten
function shootBullet() {
  const currentTime = Date.now(); // haalt de huidige tijd op
  if (currentTime - lastShotTime < SHOT_DELAY) {
    return;
  }
  /* als er minder dan 500ms voorbij zijn sinds het laatste schot
  stopt de functie die voorkomt dat je niet snel achter elkaar kunt schieten
  en kan je weer een bullet afvoeren */
  lastShotTime = currentTime;

  const bullet = document.createElement('div');
  bullet.classList.add('bullet');
  // een div wordt aangemaakt en opgeslagen in de variabele bullet, en krijgt de bullet css stijl
  bullet.style.left = PLAYERPOSITION + 'px';
  // de horizontale positie van de bullet is gebaseerd op waar de player staat
  bullet.style.bottom = '50px';
  gameContainer.appendChild(bullet);
  /* voegt de bullet toe in de gameContainer
  voegt het element toe als een sub-element */
  BULLETARRAY.push(bullet);
  // houdt elk geschoten bullet bij en slaat deze op in een array
```

```
// maakt vijanden ("invaders")
function createInvaders(level) { // (level) geeft het huidige level door en bepaald hoeveel invaders er moeten komen
  for (let i = 0; i < level * 5; i++) { // aantal invaders is afhankelijk van het level (komen +5 bij elk level)
    const invader = document.createElement('div');
    invader.classList.add('invader');
    const row = Math.floor(i / 5); // bepaalt de rij van de vijand (5 vijanden = 1 rij)
    const col = i % 5; // bepaalt de kolom van de vijand (elke rij heeft er 5)
    invader.style.left = (col * 60) + 'px';
    invader.style.top = (50 + row * 50) + 'px';
    invadersContainer.appendChild(invader); // voegt de invaders toe aan invadersContainer
    INVADERSARRAY.push(invader); // voegt de invaders toe aan de invaders array
  }
}
```

(Eggstermination)

Console Logs

```
    console.log('BEWOGEN NAAR:', PLAYERPOSITION);
    // in de console wordt vermeld op welke positie de player dan op het moment staat
} else if (e.key === 'ArrowRight' && PLAYERPOSITION < 550) {
    PLAYERPOSITION += 10;
    console.log('BEWOGEN NAAR:', PLAYERPOSITION);
} else if (e.key === ' ') {
    shootBullet();
}
```

```
if (INVADERSARRAY.length === 0) {
    level++; // als er geen invaders meer zijn, verhoogt het level met 1
    createInvaders(level);
    console.log('LEVEL:', level);
}
```

(Eggstermination)

If Else

```
if (e.key === 'ArrowLeft' && PLAYERPOSITION > 0) {
    PLAYERPOSITION -= 10;
    /* als jij (de player) het pijltje naar links indrukt en niet al helemaal tegen de muur staat,
    dan verplaats je 10px naar links */
    console.log('BEWOGEN NAAR:', PLAYERPOSITION);
    // in de console wordt vermeld op welke positie de player dan op het moment staat
} else if (e.key === 'ArrowRight' && PLAYERPOSITION < 550) {
    PLAYERPOSITION += 10;
    console.log('BEWOGEN NAAR:', PLAYERPOSITION);
} else if (e.key === ' ') {
    shootBullet();
    // als de spatiebalk wordt ingedrukt, wordt er een bullet afgeschoten
    // ' ' laat in code zien dat het om een spatiebalk gaat
}
player.style.left = PLAYERPOSITION + 'px';
```

(Eggstermination)

Functions

```
function restartGame() {
    isGameOver = false;
    score = -1; // de score begint op -1, want als je op de Game Over knop klikt telt deze klik ook als een jump
    // hierdoor zal deze klik ervoor zorgen dat de score recht voor de game begint van -1 naar 0 switched
    // de player ziet dit niet gebeuren en zal alleen merken dat de game gewoon op een score van 0 begint
    scoreElement.innerText = "Score " + score;
    block.style.animation = "block is infinite linear"; // start de animatie en loop van de blokken (eieren)
    block.style.display = "block";
    GAMEOVER.style.display = "none";
    character.style.top = "600px"; // reset de positie van de player
    block.style.left = "1480px"; // reset de positie van de eieren
    checkDead = setInterval(function() { // zie uitleg van checkDead in het vorige paragraaf
        var characterTop = parseInt(window.getComputedStyle(character).getPropertyValue("top"));
        var blockLeft = parseInt(window.getComputedStyle(block).getPropertyValue("left"));
        if (blockLeft < 150 && characterTop >= 600) {
            block.style.animation = "none";
            block.style.display = "none";
            GAMEOVER.style.display = "block";
            isGameOver = true;
            GAMEOVERSound.play();

            if (score > highScore) {
                highScore = score;
                localStorage.setItem('highScore', highScore);
                highScoreElement.innerText = "High Score: " + highScore;
            }

            clearInterval(checkDead);
        }
    }, 10);
}
```

(T-Rex Escape)

```
// maakt vijanden ("invaders")
function createInvaders(level) { // (level) geeft het huidige level door en bepaald hoeveel invaders er moeten komen
  for (let i = 0; i < level * 5; i++) { // aantal invaders is afhankelijk van het level (komen +5 bij elk level)
    const invader = document.createElement('div');
    invader.classList.add('invader');
    const row = Math.floor(i / 5); // bepaalt de rij van de vijand (5 vijanden = 1 rij)
    const col = i % 5; // bepaalt de kolom van de vijand (elke rij heeft er 5)
    invader.style.left = (col * 60) + 'px';
    invader.style.top = (50 + row * 50) + 'px';
    invadersContainer.appendChild(invader); // voegt de invaders toe aan invadersContainer
    INVADERSARRAY.push(invader); // voegt de invaders toe aan de invaders array
  }
}
```

(Eggstermination)

Arrays

```
function moveBullets() {
  BULLETARRAY.forEach((bullet, index) => {
    let bulletBottom = parseInt(bullet.style.bottom); // verticale positie van de bullet
    bullet.style.bottom = bulletBottom + 20 + 'px';
    if (bulletBottom > 600) {
      bullet.remove();
      // als de kogel de bovenkant van het scherm raakt (600), wordt hij verwijderd uit de gamecontainer
    }
  });
}
```

```
// functie die controleert of de speler heeft verloren
function checkLossCondition() {
  const playerRect = player.getBoundingClientRect();
  INVADERSARRAY.forEach(invader => {
    const invaderRect = invader.getBoundingClientRect();
    // controleert of de speler de vijand raakt
  });
}
```

(Eggstermination)

While Loops

```
if (shouldChangeDirection) {
  INVADERSDIRECTION *= -1;
  // als ^ true is, moet de richting veranderen
  let i = 0;
  while (i < INVADERSARRAY.length) {
    // na de richting veranderen, begint er een while loop die kijkt naar de locatie van de invaders
    let invader = INVADERSARRAY[i];
    invader.style.top = parseInt(invader.style.top) + 20 + 'px';
    i++;
  }
  /* de while loop gaat door, totdat i gelijk is aan INVADERSARRAY.length
  dan eindigt de lus en begint deze na de richting verandering weer opnieuw */
}
```

(Eggstermination)

For Loops

```
for (let col = 0; col < 5; col++) { // for loop die 5x loopt, een keer per kolom van de bingo kaart (de bingo kaart is natuurlijk 5x5)
  const column = document.createElement("div");
  column.classList.add("bingo-column");
  const colNumbers = shuffledNumbers.slice(col * 15, col * 15 + 15);
  // krijgt 15 getallen van de shuffledNumbers die bij de kolom passen (dus de eerste kolom (index 1) krijgt getallen tussen de 1 en 15 etc)
  if (col === 2) {
    colNumbers.splice(2, 1, 'x');
    /* het midden van de bingokaart (derde kolom, derde getal (index 2)), dan komt hier een x
    deze houdt in dat dit een FREE SPACE is en dat je deze niet hoeft in te vullen om bingo te kunnen krijgen */
  }
  for (let i = 0; i < 5; i++) { // nu een loop die 5x loopt per rij (horizontaal)
    const cell = document.createElement("div");
    cell.classList.add("bingo-cell");
    if (col === 2 && i === 2) {
      cell.classList.add("center");
      // als ze weer bij het midden komen, oftewel index 2, wordt er gemarkeert met de CSS class 'center' dat dit een FREE SPACE is
    }
    cell.textContent = colNumbers[i]; // de plekken op de bingo kaart worden ingevuld met de gekozen getallen
    cell.addEventListener("click", function() {
      this.classList.toggle("marked");
    });
    // CSS 'marked' wordt toegevoegd aan een vakje als deze aangeklikt wordt. deze kan je aan en uit togglen per klik
    column.appendChild(cell); // cell (sub-element) wordt toegevoegd aan de column (HTML-element)
  }
  bingoCard.appendChild(column);
}
```

(Bingo)

```
for (let i = 0; i < level * 5; i++) { // aantal invaders is afhankelijk van het level (komen +5 bij elk level)
  const invader = document.createElement('div');
  invader.classList.add('invader');
  const row = Math.floor(i / 5); // bepaalt de rij van de vijand (5 vijanden = 1 rij)
  const col = i % 5; // bepaalt de kolom van de vijand (elke rij heeft er 5)
  invader.style.left = (col * 60) + 'px';
  invader.style.top = (50 + row * 50) + 'px';
  invadersContainer.appendChild(invader); // voegt de invaders toe aan invadersContainer
  INVADERSARRAY.push(invader); // voegt de invaders toe aan de invaders array
}
}
```

(Eggstermination)

Events/DOM Objecten

```
// beweging van de player
document.addEventListener('keydown', (e) => {
  if (e.key === 'ArrowLeft' && PLAYERPOSITION > 0) {
    PLAYERPOSITION -= 10;
    /* als jij (de player) het pijltje naar links indrukt en niet al helemaal tegen de muur staat,
    dan verplaats je 10px naar links */
    console.log('BEWOGEN NAAR:', PLAYERPOSITION);
    // in de console wordt vermeld op welke positie de player dan op het moment staat
  } else if (e.key === 'ArrowRight' && PLAYERPOSITION < 550) {
    PLAYERPOSITION += 10;
    console.log('BEWOGEN NAAR:', PLAYERPOSITION);
  } else if (e.key === ' ') {
    shootBullet();
    // als de spatiebalk wordt ingedrukt, wordt er een bullet afgeschoten
    // ' ' laat in code zien dat het om een spatiebalk gaat
  }
  player.style.left = PLAYERPOSITION + 'px';
});
```

(Eggstermination)

```
document.addEventListener('keydown', function(event) { // checkt of er een knop wordt ingedrukt
    if (event.code === 'Space') { // checkt of de spatiebalk wordt ingedrukt
        jump();
    }
});
```

(T-Rex Escape)